User's Manual

RENESAS

CONFIDENTIAL

R-Car Gen4 AUTOSAR R19-11 MCAL

User's Manual

Getting Started

RTM8RC79FRCMCL5DA0JCDRE, RTM8RC79FRCMCL5QA0JCDRE
RTM8RC79FRCSPL5DA0JCDRE, RTM8RC79FRCSPL5QA0JCDRE
RTM8RC79FRCCOM5DA0JCDRE, RTM8RC79FRCCOM5QA0JCDRE
RTM8RC79FGCMCL5DB0JCDRE, RTM8RC79FGCMCL5QB0JCDRE
RTM8RC79FGCSPL5DB0JCDRE, RTM8RC79FGCSPL5QB0JCDRE
RTM8RC79FGCCOM5DB0JCDRE, RTM8RC79FGCCOM5QB0JCDRE
RTM8RC779GCMCL5DA0JCDRE, RTM8RC779GCMCL5QA0JCDRE
RTM8RC779GCSPL5DA0JCDRE, RTM8RC779GCSPL5QA0JCDRE
RTM8RC779GCCOM5DA0JCDRE, RTM8RC779GCCOM5QA0JCDRE
RTM8RC779GCCDD5DA0JCDRE, RTM8RC779GCCDD5QA0JCDRE
RTM8RC779HCMCL5DA0JCDRE, RTM8RC779HCMCL5QA0JCDRE
RTM8RC779HCSPL5DA0JCDRE, RTM8RC779HCSPL5QA0JCDRE
RTM8RC779HCCOM5DA0JCDRE, RTM8RC779HCCOM5QA0JCDRE
RTM8RC779HCCDD5DA0JCDRE, RTM8RC779HCCDD5QA0JCDRE

Target Device:
R-Car S4 (Includes S4N)
R-Car V4H
R-Car V4M

Renesas Electronics
www.renesas.com

Rev.2.01 Oct. 2023

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.

List of Abbreviations and Acronyms

Abbreviation	Full Form
ARXML/arxml	AUTOSAR xml
AUTOSAR	Automotive Open System Architecture
BSWMDT	Basic Software Module Description Template
<MSN>	Module Short Name
<msn>	module short name
ECU	Electronic Control Unit
GUI	Graphical User Interface
MB	Mega Bytes
MHz	Mega Hertz
RAM	Random Access Memory
xml/XML	eXtensible Markup Language
<DOMAIN_NAME>	RCAR, RH850
<MICRO_VARIANT>	S4, V4H, V4M
<MICRO_SUB_VARIANT>	S4, S4N, V4H, V4M
<AUTOSAR_VERSION>	4.5.0
DEVICE_NAME	RTM8RC79FR RTM8RC79FG V4H V4M

Definitions

Term	Represented by
.xml	XML File.
.arxml	AUTOSAR XML File.
.trxml	Translation XML File.
ECU Configuration Parameter Definition File	The ECU Configuration Parameter Definition is of type XML that contains the definition for AUTOSAR software components, i.e. definitions for Modules, Containers and Parameters. The format of the XML File will be compliant with AUTOSAR ECU specification standards.
ECU Configuration Description File	The ECU Configuration Description file in XML format that contains the configured values for Parameters, Containers and Modules. ECU Configuration Description XML File format will be compliant with the AUTOSAR ECU specification standards.
BSWMDT File	The BSWMDT File in XML format that is the template for the Basic Software Module Description. BSWMDT File format will be compliant with the AUTOSAR BSWMDT specification standards.
Translation XML File	Translation XML File is in XML format that contains translation and device-specific header file path.
Configuration XML File	Configuration XML File is in XML format that contains command line options and options for input/output file path.

Table of Contents

1. Introduction 5

2. Generation Tool 6
2.1 Generation Tool of modules using Perl 6
2.1.1 Usage 6
2.1.2 Sample Usage 9
2.2 Generation Tool of modules using C Sharp 11
2.2.1 Installation Requirements 12
2.2.1.1 Hardware Requirements 12
2.2.1.2 Software Requirements 12
2.2.1.3 Limitations 12
2.2.2 Installation 12
2.2.2.1 Prerequisite 12
2.2.2.2 Installation Steps 12
2.2.3 Uninstallation 12
2.2.4 Usage 12
2.2.5 Sample Usage 14
2.3 ECU Configuration Description File 16
2.4 BSWMDT File 17
2.5 Translation XML File 18
2.5.1 Translation Header File 18
2.5.2 Device-Specific Header File 18
2.6 Configuration XML File 19
2.7 Error Messages 19
2.8 Warning Messages 24
2.9 Information Messages 25

3. Application Example 26
3.1 Folder Structure 26
3.2 Makefile Description 26

4. Support for Different Interrupt Categories 28
4.1 The <MSN> Driver supports CAT1 and CAT2 interrupt categories: 28
4.2 Using MCAL integrated with Operation System: 28
4.3 Using MCAL stand alone 28

5. GNU MAKE Environment 30
5.1 Build Process Without GNUMAKE 30
5.2 Build Process With GNUMAKE 34

6. Load Binaries 35

List of Figures

Figure 2.1 Generation Tool Overview 6
Figure 2.2 Generation Tool Deployment 11
Figure 2.3 Flow of File Generation for Not Multi Instance 16
Figure 5.1 How to Open the System Window 30
Figure 5.2 How to Open the Advanced System Setting Window 31
Figure 5.3 How to Open Environment Variables Window 32
Figure 5.4 Environment Variables Window 33
Figure 5.5 How to Set Environment Variables 34

List of Tables

Table 2-1 Files that deviate from the naming conventions 6
Table 2-2 Mandatory Parameters 21
Table 3-1 Makefile description 26
Table 4-1 CAT1 and CAT2 Naming Convention 28
Table 4-2 List of ISR Names that need to be configured and published in Os.h (CAT2) or used in the interrupt vector table (CAT1) for <MSN> Driver 28

1.Introduction

The document describes the information about installation, Generation Tool, and the references to Sections in the Component User Manuals that the user needs to refer to build the executable.

Generation Tool is a command line tool that accepts ECU Configuration Description File(s), BSWMDT File, Translation XML File and Configuration XML File as input, and generates the C source and header files based on the configuration of the module.

2.Generation Tool

Generation Tool is a command line tool that provides scalability and configurability for the component. It accepts ECU Configuration Description File(s), BSWMDT File, Translation XML File and Configuration XML File as input, and generates the C Header and C Source files. However, Configuration XML File is optional.

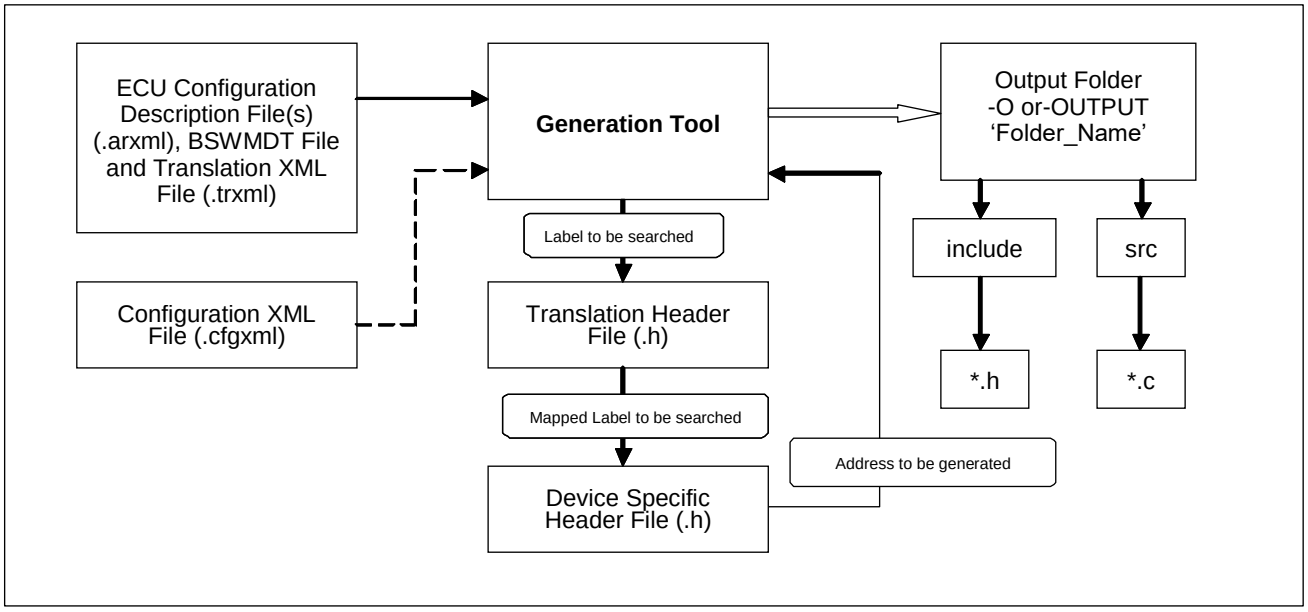
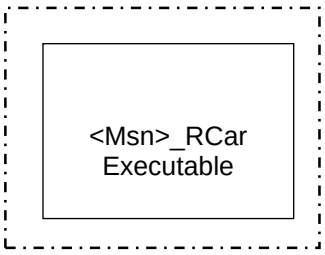


Figure 2.1 Generation Tool Overview

2.1 Generation Tool of modules using Perl

The Generation Tool for modules Driver Component using Perl language is located at below location:
rel\modules\<msn>\generator\<Msn>RCar.exe



Note:
In Ethernet Driver (S4_CR52), some of the files deviate from the naming conventions described in this document.

Table 2-1 Files that deviate from the naming conventions

Convention	Actual file name
<Msn>_PBcfg.c	Eth_Rsw2_PBcfg.c
<Msn>_Cfg.h	Eth_Rsw2_Cfg.h

2.1.1 Usage

This section provides the information regarding the usage of the Generation Tool. It also provides the syntax of the command line arguments (input filenames and options).

Generation Tool executable is invoked as shown below.

{<Msn>_RCar.exe} <Options> <Input Filename(s)>

Where,

<Msn>_RCar.exe: Name of the Generation Tool Executable

Options: [-H/-Help -C/-Configfile -O/-Output -Osrc -Oinc -L/-Log
-D/-Dryrun/-Fr/Filter]

Input Filename(s): {ECU Configuration Description File(s), BSWMDT File and
Translation XML File [optional]}

Option: -H/-Help
To display help on using the tool. Gets the highest priority when used with other options.

Option: -C/-Configfile
To execute tool with the options provided in the Configuration XML File. Command line options get the higher priority than the options provided in Configuration XML File.

Option: -O/-Output
By default, the tool generates output files in the
'<Msn>_Output' folder in the path where executable is present. The user can use the -O option followed by the folder name, to generate the output files in the specified folder.
The C Source and C Header files are generated in the sub folders 'src' and 'include' within the output folder.

Option: -Osrc
The user can use the -Osrc option followed by the folder name to generate the C Source files in the specified folder.

Option: -Oinc
The user can use the -Oinc option followed by the folder name to generate the C Header files in the specified folder.

Option: –L/-Log
To log the output to the <Msn>.log file.
If this option is not provided, output folder specified by -O/-Output is used.

Option: -D/-Dryrun
To execute tool in validation mode. The tool will not generate output files even though the input file provided is error-free.

Option: -Fr/-Filter
To select a vendor by name. ECU Configuration Description File(s) may have configurations of more than one vendor, using the option to select the expected vendor. "Renesas" is used by default.

Note

- If Translation XML File is not provided on the command line, then '<Msn>.trxml' present in the same location as '<Msn>_RCar.exe' is considered as 'default' Translation XML File by the Generation Tool.
- If Configuration XML File is not provided on the command line, then '<Msn>.cfgxml' present in the same location as '<Msn>_RCar.exe' is considered as 'default' Configuration XML File by the Generation Tool.
- The Generation Tool should not be executed more than five times in parallel
- At the end of Generation Tool execution, it will return value 1 if there are any errors, otherwise it returns 0. This feature should be used for automation testing to validate pass/fail result of execution.

- Either absolute path or relative path can be provided to specify the location of folder or file.
- If Output folder specified by -O/-Output is provided as invalid, the Log file cannot be opened.

2.1.2 Sample Usage

Sample usage of the generation tool is given below. "<Msn>_RCar.exe" is taken as example. Similar usage is applicable for other MCAL Generation Tools.

<Msn>_RCar.exe

Generation Tool usage is displayed on the terminal. -M/Module option is required to specify the MCAL Driver module used to generate C Header and C Source output files.

<Msn>_RCar.exe -H

Displays Generation Tool help information on the terminal where Generation Tool executable is present.

<Msn>_RCar.exe -L -O output Sample.arxml BSWMDT.arxml

Generation Tool logs the output to the <Msn>.log file. <Msn>_PBcfg.c file is generated in 'src' folder. <Msn>_Cfg.h file is generated in 'include' folder.

<Msn>_RCar.exe -D Sample.arxml BSWMDT.arxml

Generation Tool validates an input file and displays error/warning/information messages on the command line, if any. Output files are not generated since –D option is provided in the command line.

<Msn>_RCar.exe -O output Sample.arxml BSWMDT.arxml

Output files are generated in "output" folder, <Msn>_PBcfg.c in 'src' folder, and <Msn>_Cfg.h file in 'include' folder.

<Msn>_RCar.exe C:\Input\Sample.arxml C:\Input\BSWMDT.arxml -O output

Generation Tool accepts input file (Sample.arxml) from absolute directory path "C:\Input". Output files are generated in folder "output". <Msn>_PBcfg.c is generated in 'src' folder, and <Msn>_Cfg.h file in 'include' folder.

<Msn>_RCar.exe Sample.arxml BSWMDT.arxml -O C:\Output

Output files are generated in folder "C:\Output", <Msn>_PBcfg.c in 'src' folder, and <Msn>_Cfg.h file in 'include' folder.

<Msn>_RCar.exe Sample.arxml BSWMDT.arxml Sample.trxml

Generation Tool accepts ECU Configuration Description File (Sample.arxml), BSWMDT File (BSWMDT.arxml) and Translation XML File (Sample.trxml) from the current working directory. Output files are generated in the default folder "<Msn>_Output", since –O option is not provided in the command line.
<Msn>_PBcfg.c is generated in 'src' folder, and <Msn>_Cfg.h file in 'include' folder.

<Msn>_RCar.exe -C Sample.cfgxml Sample.arxml BSWMDT.arxml Sample.trxml

Generation Tool accepts ECU Configuration Description File (Sample.arxml), BSWMDT File (BSWMDT.arxml) and Configuration XML File (Sample.cfgxml) from the current working directory. Tool accepts options provided in the Configuration XML File. If the Configuration XML File name is not provided as input with -C option, Generation Tool fails.

<Msn>_RCar.exe –FR Renesas

Only Renesas configurations will be generated.

Note

- If Translation XML File is not provided on the command line,
  <Msn>_RCar.exe considers <Msn>.trxml as 'default' Translation XML File from the same directory where the tool is located.

- If Configuration XML File is not provided on the command line,
  <Msn>_RCar.exe considers <Msn>.cfgxml as 'default' Configuration XML File from the same directory where the tool is located.

- If any filename/directory name related argument on the command line contain the 'space', then the same argument on the command line should be provided in double quotes "" as followed by standard command line feature. E.g., if

the file name is ‘Sample Description.arxml’, then on the command line, the same name should be provided in double quotes “” as “Sample Description.arxml”.

- The ‘include’ and ‘src’ directories are generated inside the output directory provided on the command line or in the default output directory <Msn>_Output.
- BSWMDT file should not be updated manually since it is “Static Configuration” file.

Note

None

2.2 Generation Tool of modules using C Sharp

Generation Tool comprises 2 components, MCALConfGen executable and library files (dlls) for each module, e.g. CanRCAR.dll, CanRH850.dll. At runtime, the executable loads the library files to generate output files.

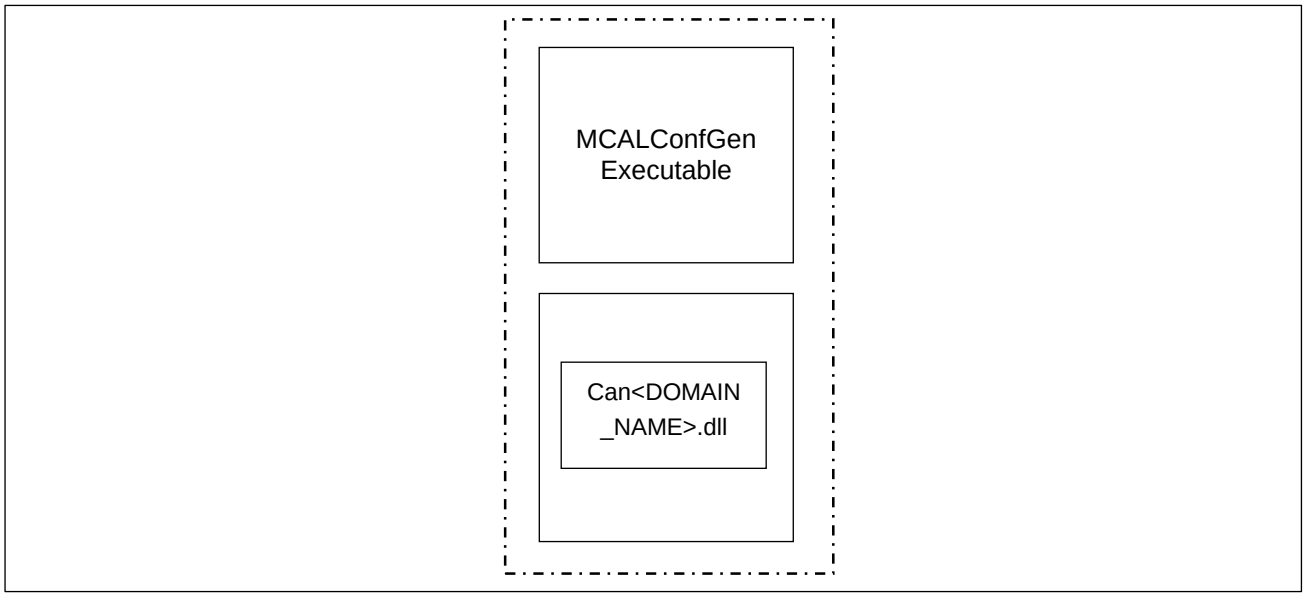


Figure 2.2 Generation Tool Deployment

2.2.1 Installation Requirements

The minimum hardware and software requirements for proper installation of Generation Tool are listed below. This ensures optimal performance of the tool.

2.2.1.1 Hardware Requirements

Processor	Pentium/equivalent processor @ 1 GHz or greater
Memory	RAM 2 GB or greater
Hard Disk Drive	20 GB or greater storage capacity

2.2.1.2 Software Requirements

Operating System	Microsoft Windows 10
.NET Framework	4.5

2.2.1.3 Limitations

Command Line characters are limited to 1024 depending on the operating system.

2.2.2 Installation

The installation procedure of the Generation Tool is provided in the section below.

2.2.2.1 Prerequisite

Generation Tool executable runs on Windows platforms only.

2.2.2.2 Installation Steps

Copy the Generation Tool executable file to the local hard disk.

Copy DLLs file of the specific modules to directory dlls/<MICRO_VARIANT> from the same directory where the tool is located. Naming rule of module-specific DLL is <Msn><MICRO_VARIANT>.dll

Run the executable with -H option to get help on usage of the tool.

MCALConfGen.exe -H

This command generates the usage of Module-Specific Driver Generation Tool on the command line.

2.2.3 Uninstallation

There is no specific method for uninstalling the Generation Tool. To uninstall, delete the Generation Tool executable and dlls directory from the existing directory.

2.2.4 Usage

This section provides the information regarding the usage of the Generation Tool. It also provides the syntax of the command line arguments (input filenames and options).

Generation Tool executable is invoked as shown below.

{MCALConfGen.exe} <Options> <Input Filename(s)>

Where,

MCALConfGen.exe: Name of the Generation Tool Executable

Options: [-H/-Help -M/Module -C/-Configfile -O/-Output -Osrc -Oinc -L/-Log -D/-Dryrun/-Fr/Filter]

Input Filename(s): {ECU Configuration Description File(s), BSWMDT File and Translation XML File [optional]}

Option: -H/-Help

To display help on using the tool. Gets the highest priority when used with other options.

Option: -M/-Module

To specify MCAL Driver module used to generate C Header and C Source output files. Msn of modules will be used as an input value of the option.

Option: -C/-Configfile

To execute tool with the options provided in the Configuration XML File. Command line options get the higher priority than the options provided in Configuration XML File.

Option: -O/-Output

By default, the tool generates output files in the

'<Msn>_Output' folder in the path where executable is present. The user can use the -O option followed by the folder name, to generate the output files in the specified folder.

The C Source and C Header files are generated in the sub folders 'src' and 'include' within the output folder.

Option: -Osrc

The user can use the -Osrc option followed by the folder name to generate the C Source files in the specified folder.

Option: -Oinc

The user can use the -Oinc option followed by the folder name to generate the C Header files in the specified folder.

Option: –L/-Log

To log the output to the <Msn>.log file.

If this option is not provided, output folder specified by -O/-Output is used.

Option: -D/-Dryrun

To execute tool in validation mode. The tool will not generate output files even though the input file provided is error-free.

Option: -Fr/-Filter

To select a vendor by name. ECU Configuration Description File(s) may have configurations of more than one vendor, using the option to select the expected vendor. "Renesas" is used by default.

Note

- If Translation XML File is not provided on the command line, then '<Msn>.trxml' present in the same location as 'MCALConfGen.exe' is considered as 'default' Translation XML File by the Generation Tool.
- If Configuration XML File is not provided on the command line, then '<Msn>.cfgxml' present in the same location as 'MCALConfGen.exe' is considered as 'default' Configuration XML File by the Generation Tool.
- The Generation Tool should not be executed more than five times in parallel
- At the end of Generation Tool execution, it will return value 1 if there are any errors, otherwise it returns 0. This feature should be used for automation testing to validate pass/fail result of execution.
- Either absolute path or relative path can be provided to specify the location of folder or file.

- If Output folder specified by -O/-Output is provided as invalid, the Log file cannot be opened.

2.2.5 Sample Usage

Sample usage of the generation tool is given below. “MCALConfGen.exe” is taken as example. Similar usage is applicable for other MCAL Generation Tools.

MCALConfGen

Generation Tool usage is displayed on the terminal. -M/Module option is required to specify the MCAL Driver module used to generate C Header and C Source output files.

MCALConfGen -M <Msn>

Generation Tool usage is displayed on the terminal. It accepts Configuration XML File as default and performs the execution, based on the settings provided in Configuration XML File.

MCALConfGen -H

Displays Generation Tool help information on the terminal where Generation Tool executable is present.

MCALConfGen -M <Msn> -L -O output Sample.arxml BSWMDT.arxml

Generation Tool logs the output to the <Msn>.log file. <Msn>_PBcfg.c file is generated in ‘src’ folder. <Msn>_Cfg.h file is generated in ‘include’ folder.

MCALConfGen -M <Msn> -D Sample.arxml BSWMDT.arxml

Generation Tool validates an input file and displays error/warning/information messages on the command line, if any. Output files are not generated since –D option is provided in the command line.

MCALConfGen -M <Msn> -O output Sample.arxml BSWMDT.arxml

Output files are generated in “output” folder, <Msn>_PBcfg.c in ‘src’ folder, and <Msn>_Cfg.h file in ‘include’ folder.

MCALConfGen -M <Msn> C:\Input\Sample.arxml C:\Input\BSWMDT.arxml -O output

Generation Tool accepts input file (Sample.arxml) from absolute directory path “C:\Input”. Output files are generated in folder “output”. <Msn>_PBcfg.c is generated in ‘src’ folder, and <Msn>_Cfg.h file in ‘include’ folder.

MCALConfGen -M <Msn> Sample.arxml BSWMDT.arxml -O C:\Output

Output files are generated in folder “C:\Output”, <Msn>_PBcfg.c in ‘src’ folder, and <Msn>_Cfg.h file in ‘include’ folder.

MCALConfGen -M <Msn> Sample.arxml BSWMDT.arxml Sample.trxml

Generation Tool accepts ECU Configuration Description File (Sample.arxml), BSWMDT File (BSWMDT.arxml) and Translation XML File (Sample.trxml) from the current working directory. Output files are generated in the default folder “<Msn>_Output”, since –O option is not provided in the command line.
<Msn>_PBcfg.c is generated in ‘src’ folder, and <Msn>_Cfg.h file in ‘include’ folder.

MCALConfGen -M <Msn> -C Sample.cfgxml Sample.arxml BSWMDT.arxml Sample.trxml

Generation Tool accepts ECU Configuration Description File (Sample.arxml), BSWMDT File (BSWMDT.arxml) and Configuration XML File (Sample.cfgxml) from the current working directory. Tool accepts options provided in the Configuration XML File. If the Configuration XML File name is not provided as input with -C option, Generation Tool fails.

MCALConfGen -M <Msn> –FR Renesas

Only Renesas configurations will be generated.

Note

- If Translation XML File is not provided on the command line,
  MCALConfGen.exe -M <Msn> considers <Msn>.trxml as ‘default’ Translation XML File from the same directory where the tool is located.

- If Configuration XML File is not provided on the command line,
  MCALConfGen.exe -M <Msn> considers <Msn>.cfgxml as ‘default’ Configuration XML File from the same directory where the tool is located.

- If any filename/directory name related argument on the command line contain the ‘space’, then the same argument on the command line should be provided in double quotes “” as followed by standard command line feature. E.g. if the file name is ‘Sample Description.arxml’, then on the command line, the same name should be provided in double quotes “” as “Sample Description.arxml”.

- The ‘include’ and ‘src’ directories are generated inside the output directory provided on the command line or in the default output directory <Msn>_Output.

- BSWMDT file should not be updated manually since it is “Static Configuration” file.

Note:

<msn>: dio, port, can, eth, gpt, spi, wdg, mcu, fls, cddiccom, cddemm, cddcrc, cddths, cddiic, cddrfso, cddipmmu.

<Msn>: Dio, Port, Can, Eth, Gpt, Spi, Wdg, Mcu, Fls, Cddiccom, Cddemm, Cddcrc, Cddths, Cddiic, Cddrfso, Cddipmmu.

2.3 ECU Configuration Description File

This holds the configuration values for parameters, containers, modules and references to other module's configurations. Using the information to generate configuration code of basic software modules. An ECU Configuration Description File can contain configuration information of one or more modules.
The Generation Tool supports multi-instance. (not supported S4, V4H, V4M)
If the Configuration Description File has only one configuration set, the Generation Tool does not add any postfix.

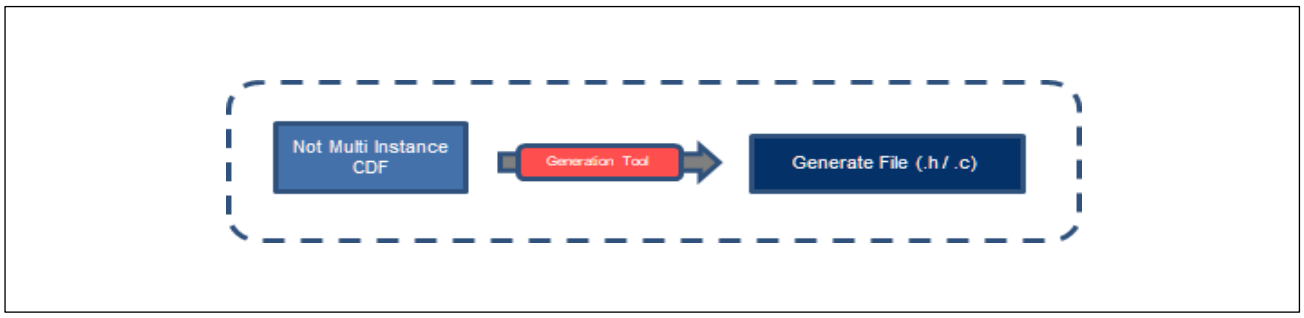


Figure 2.3 Flow of File Generation for Not Multi Instance

2.4 BSWMDT File

The BSWMDT File is the template for the Basic Software Module Description. Generation Tool uses “Common Published Information” from module specific BSWMDT file. This file should not be updated manually since it is “Static Configuration” file.

The required elements of the BSWMDT File by Generation Tool are as follows:

- BSW-MODULE-DESCRIPTION
- MODULE-ID
- BSW-IMPLEMENTATION
- SW-VERSION
- VENDOR-ID
- AR-RELEASE-VERSION
- VENDOR-SPECIFIC-MODULE-DEF-REFS

In case of multiple driver support implementation, VENDOR-API-INFIX is mandatory, but not used for the single driver support implementation.

2.5 Translation XML File

Generation Tool accepts ECU Configuration Description File(s) (.arxml), BSWMDT File (.arxml) and Translation XML File (.trxml) as an input. Translation XML File is in XML format which contains translation and device-specific header file path.

If the mapped device-specific address label is changed/updated, then, only the respective mapping in Translation Header File needs to be updated. In this case, there will not be any impact on Generation Tool to generate the address in tool-generated output file(s).

Note Generation Tool will work with only last Translation XML File in case user provides multiple Translation XML File path via command line or Configuration XML File.

2.5.1 Translation Header File

This file is look-up table (mapping in the form of definitions) for the device-specific address labels. Based on the configuration in ECU Configuration Description File, the labels will be searched in Device-Specific Header File by Generation Tool to generate the associated address in tool-generated output file(s). For the Translation Header File path, the value of '<Msn>DeviceName' parameter in the container '<Msn>General' should be present as child tag of TRANSLATION- FILE-PATH in Translation XML File. Both 'Absolute' and 'Relative' paths are supported by generation tool, however the default path is 'Relative' path.

E.g.

<TRANSLATION-FILE-PATH>

<Value_Of_MsnDeviceName>..\DF_Timer.h ..\DF_Timer_ISR.h</ Value_Of_MsnDeviceName>

</TRANSLATION-FILE-PATH>

2.5.2 Device-Specific Header File

This file contains device-specific labels and the associated address. Based on the configuration in ECU Configuration Description File, the mapped device-specific address labels will be used to generate the associated address in tool-generated output file(s). For the Device-Specific Header File path, the value of '<Msn>DeviceName' parameter in the container '<Msn>General' should be present as child tag of DEVICE-FILE-PATH in Translation XML File. Both 'Absolute' and 'Relative' paths are supported by generation tool, however default path is 'Relative' path.

If multiple Device-Specific Header Files needs to be provided for the same device (value of '<Msn>DeviceName' parameter) in Translation XML File, then each Device-Specific Header File path should be separated with 'space'.

E.g.

<DEVICE-FILE-PATH>

<Value_Of_MsnDeviceName>..\DF_Timer.h ..\DF_Timer_ISR.h</ Value_Of_MsnDeviceName>

</DEVICE-FILE-PATH>

Note Generation Tool will search the mapped labels in Device-Specific Header File by using Translation Header File for the respective address generation in tool-generated output file(s).

2.6 Configuration XML File

Configuration XML File is in the XML format which contains command line options and input/output path. For the syntax of the contents of Configuration XML File, refer to the Chapter 7 Appendix.

E.g.

<LOG>ON/OFF</LOG>

<HELP>ON/OFF</HELP>

The contents of Configuration XML file are overridden by Command line options. Refer to the Section 2.1.1 Usage.

Note Either absolute path or relative path can be provided to specify the location of folder or file.

2.7 Error Messages

This section contains the list of error/warning/information messages common to AUTOSAR Renesas R19-11 MCAL Driver module that will be generated by the Generation Tool.

ERR000001: File <File_Name> does not exist.

This error occurs, if the input <File_Name> is not found.

ERR000002: Name of the Generation Tool configuration XML File is not given along with the <-C/-CONFIGFILE>.

This error will occur, if the name of the Generation Tool configuration XML file is not given along with the -C/-CONFIGFILE option.

ERR000004: Cannot open the <Log file name> file.

This error will occur, if unable to open the <Log file name> file.

ERR000005: Name of output directory is not given along with <input option> option.

This error will occur, if the output directory name is not given along with -O/-OUTPUT, -Osrc or -Oinc options.

ERR000006: Name of output directory is not given in OUTPUT-PATH tag in <File name>.

This error will occur, if the output directory is not given in OUTPUT-PATH tag in configuration file.

ERR000007: The Generation Tool expects inputs.

This error will occur, if no option is provided in the command line and none of the option in the configuration file is set.

ERR000008: The option <option> is not supported by the Generation Tool. The Generation Tool supports <-O/-OUTPUT, -Osrc, -Oinc, -H/-HELP, -L/-LOG, -C/-CONFIGFILE and -D/-DRYRUN>" options.

This error will occur, if the invalid <option> is provided to the tool.

ERR000009: Invalid output directory name <output directory name> as the file with same name exists.

This error will occur, if the <output directory name> already exists.

ERR000010: Invalid output directory name <output directory name> Directory name should not contain any of *\?\"\|\: characters.

This error will occur, if the <output directory name> path contains junk character.

ERR000011: ECU Configuration Description File is not provided as input to the Generation Tool.

This error will occur, if the ECU Configuration Description File is not given in the command line or in configuration file.

ERR000012: The input <File name> is not as per XML standard. Provide the ECU Configuration Description File as input on the command line.

This error will occur, if the ECU Configuration Description File does not follow the XML standard.

ERR000013: <File name> should contain the TRANSLATION-FILE-PATH' and 'DEVICE-FILE-PATH' tags.

This error will occur, if the translation <File name> does not have ‘TRANSLATION-FILE-PATH’ and 'DEVICE-FILE-PATH' tags.

ERR000014: 'TRANSLATION-FILE-PATH' tag in <File name> is empty.

This error will occur, if the translation <File name> does not have ‘TRANSLATION-FILE-PATH’ tags.

ERR000015: The 'device_name' tag should be present as child of 'TRANSLATION-FILE-PATH'' tag in <File name>.

This error will occur, if the device mentioned in ECU Configuration Description File is not present in 'TRANSLATION-FILE-PATH’ tag in the <File name>.

ERR000016: ‘DEVICE-FILE-PATH’ tag in <File name> is empty.

This error will occur, if the content of ‘DEVICE-FILE-PATH’ tag in translation file is empty.

ERR000017: The 'device_name’ tag should be present as child of ‘DEVICE-FILE-PATH' tag in <File name>.

This error will occur, if the device mentioned in ECU Configuration Description File is not present in ‘DEVICE-FILE-PATH’ tag.

ERR000018: Cannot create directory <output directory name>. <Detail Message>.

This error will occur, if output directory <output directory name> cannot be created and detail message also explains the reason for the failure.

ERR000019: Cannot open <File name>.

This error will occur, if unable to open <File name>.

ERR000020: The macro label <macro label> should be unique in <translation file name> translation C Header File.

This error will occur, if macro label is not unique in translation C Header File.

ERR000021: The macro definition for <macro label> macro is not found in <translation file name> translation C Header File.

This error will occur, if macro definition is not found in translation C Header File.

ERR000022: The macro value for <macro label> macro is empty in <translation file name> translation C Header File.

This error will occur, if macro label value is empty in translation C Header File.

ERR000023: The macro definition for <macro value> macro is not found in <device file name> device specific C Header File(s).

This error will occur, if macro definition is not found in input device-specific C Header File(s).

ERR000024: The macro value for <macro value> macro is empty in <device file name> device specific C Header File(s).

This error will occur, if macro value is empty in input device-specific C Header File(s).

ERR000025: Path <Configured Reference Path> provided for Bsw Module is incorrect.

This error will occur, if the reference provided for Bsw Module Component is incorrect.

ERR000026: BSWMDT content is not present in the input file(s) for '<Module Name>' module.

This error will occur, if the module-specific BSWMDT content is not present in the input files.

ERR000028: 'MODULE-DESCRIPTION-REF' element should be present in the description file of '<Module Name>' module.

This error will occur, if the MODULE-DESCRIPTION-REF element is not present in the module-specific description file.

ERR000029: AUTOSAR version <value> of BSWMDT File is inconsistence with ECU Configuration Description File of <variant> device.

This error will occur, if the AUTOSAR version of the BSWMDT File is inconsistent with the CDF device.

ERR000030: The 'parameter tag name' tag should be configured in BSWMDT File.

This error will occur, if any of the configuration parameter(s) mentioned below is (are) not configured in BSWMDT File.

The list of mandatory parameters with respect to container is listed below:

Table 2-2 Mandatory Parameters

Container	Parameters
BswImplementation	SwVersion
	VendorId
	ArReleaseVersion
BswModuleDescription	ModuleId

ERR000031: Module name is not specified by option <-m> or NOT in supported modules <Option list>.

This error will occur, if option -m does not exist or module name is not specified by option -m.

ERR000032: Filter value is not given along with <-FR/-FILTER> option.

This error will occur, if vendor name is not specified by -FR/-FILTER option.

ERR000034: Options <Option list> should be set explicitly by command line options and configuration file.

This error will occur, if options -H/-HELP, -L/-LOG and -D/-DRYRUN are not explicitly specified by command line options and configuration file.

ERR000036: BSWMDT file is not provided as input to the Generation Tool.

This error will occur, if BSWMDT file is not provided as input to the Generation Tool.

ERR000037: The register label <register label> should be unique in < translation File name> device C Header File.

This error will occur, if one register label in the device C Header File is defined for more than one address.

ERR000038: The register address <register address> has invalid Hex value in <device file name> device C Header File.

This error will occur, if register address has invalid Hex value in the device C Header File.

ERR000039: Device name is NOT specified in CDF files.

This error will occur, if device name is NOT specified in the CDF files.

ERR000040: More than one BSWMDT files are provided as input to the Generation Tool.

This error will occur, if more than one BSWMDT files are provided as input to the Generation Tool.

ERR000041: <MSN>_E_INT_INCONSISTENT should be configured when the parameter <Msn>InterruptConsistencyCheck is true.

This error will occur, if <Msn>InterruptConsistencyCheck is enabled and <MSN>_E_INT_INCONSISTENT is not configured.

ERR000043: <Dll> file is not compatible.

This error will occur, if the loaded dll file is not compatible with the current executable file.

ERR000045: <Dll> file is not found.

This error will occur if the specific dll file is not found.

ERR000046: Incorrect device name <devicename>. Generation tool supports device name(s): <Supported devicenames>.

This error will occur, if device name is NOT supported.

ERR000047: No vendor data found from CDF files.

This error will occur, if vendor does not exist in the CDF for <MSN>.

ERR000048: Path <VENDOR-SPECIFIC-MODULE-DEF-REF> in BSWMDT is not match to path <DEFINITION-REF> in CDF files.

This error will occur, if the value of VENDOR-SPECIFIC-MODULE-DEF-REF in BSWMDT does not match the value of DEFINITION-REF in the CDF files.

ERR000049: VENDOR-SPECIFIC-MODULE-DEF-REF does not exist in BSWMDT.

This error will occur, if the value of VENDOR-SPECIFIC-MODULE-DEF-REF does not exist in the BSWMDT.

ERR000050: No permission to perform action: [read file/list files/create directory] <file/directory path>.

This error will occur, if users do not have the permission to read file/list files/create directory.

ERR000051: Path too long, cannot perform action: [read file/list files/create directory] <file/directory path>.

This error will occur, if the path of file/directory is too long.

ERR000053: An I/O error occurred while perform action [read/write] the file <file path>.

This error will occur, if there is an I/O error while performing action [read/write] on the file.

ERR000055: Unable to load file <file path>.

This error will occur, if unable to load file. Detail message also explains the reason for the failure.

ERR000056: Bad DLL format, unable to load file <filename>. <Detail Message>.

This error will occur, if unable to load file due to bad DLL format.

Note If error is detected, the Generation Tool will report the error and be aborted immediately.

2.8 Warning Messages

WRN000001: As per AUTOSAR ECU Configuration Description File naming convention, the file extension should be '.arxml' for file.

This warning will occur, if the ECU Configuration Description file has an extension other than ’.arxml’.

WRN000002: There are duplications in command lines.

This warning will occur, if there are duplicate command options.

2.9 Information Messages

INF000001: Tool Version:

This is to display Tool Version for each execution of the tool.

INF000002: Command line arguments:

This is to display the command line arguments for each execution of the tool.

INF000003: The valid inputs are provided below.

This information will occur if the command line option is not given.

INF000004: Opened file <filename> at <time>.

This information will occur, while opening the file.

INF000005: Error(s) and Warning(s) detected.

This information will display the number of errors and warnings.

INF000006: Execution completed successfully.

This information will occur, if the execution completed successfully.

INF000007: Execution completed successfully with warnings.

This information will occur, if the execution completed successfully with warnings.

INF000008: Execution terminated due to command line errors.

This information will occur, if the execution terminated due to command line errors.

INF000009: Execution terminated due to error in the input file.

This information will occur, if the execution terminated due to error in the input file.

INF000010: Execution terminated due to error, during the structure generation in the output file.

This information will occur, if the execution terminated while generating structure in output file.

INF000011: Execution terminated because file cannot be overwritten since it is read only.

This information will occur, if the output file is read only.

INF000012: Execution terminated because of incompatible dll file.

This information will occur, if the loaded dll file is not compatible.

INF000014: Execution terminated because specific dll file is not found.

This information will occur, if the specific dll file is not found.

3.Application Example

3.1 Folder Structure

Refer to "R-Car Gen4 AUTOSAR R19-11 MCAL Modules Overview User's Manual" 3.2 Folder Structure.

3.2 Makefile Description

The sample application use Makefile. This section describes those files.
The **Table 3-1** shows description of Makefile.

Table 3-1 Makefile description

Location:	Files	Description
rel\common\generic\compiler\<AUTOSAR_VERSION>\ghs\make	ghs_rh850_r<AUTOSAR_VERSION>_defs.mak	This file contains compile options.
rel\common\generic\compiler\<AUTOSAR_VERSION>\arm\make	ghs_rcar_r<AUTOSAR_VERSION>_defs.mak	This file contains compile options.
rel\<MICRO_VARIANT>\common_family\make\ghs	Common.mak	This file contains common make rules of Sample Application.
rel\<MICRO_VARIANT>\common_family\make\arm	Common.mak	This file contains common make rules of Sample Application.
rel\modules\<msn>\make	renesas_<msn>_rules.mak	This file contains module make rules of Sample Application.
rel\modules\<msn>\sample_application\make\ghs	App_<MSN>_Common_Sample.mak	This file contains flags for including items.

rel\modules\<msn>\sample_application\make\arm	App_<MSN>_Common_Sample.mak	This file contains flags for including items.

Other Makefile is used for including files which use sample application.

The make rules can use from SampleApp.bat for debugging sample application.

Location:
rel\<MICRO_VARIANT>\common_family\make\ghs\SampleApp.bat
rel\<MICRO_VARIANT>\common_family\make\arm\SampleApp.bat

Details refer to "R-Car Gen4 AUTOSAR R19-11 MCAL Modules Overview User's Manual" 3.6.2 Building Sample Application.

4.Support for Different Interrupt Categories

4.1 The <MSN> Driver supports CAT1 and CAT2 interrupt categories:

CAT1

In CAT1 type of interrupt, the ISR does not use an operating system service. In practice, the OS does not handle these interrupts, and the interrupt handler is implemented in the driver code, with the only restriction that OS services cannot be called. Typically, these are the fastest highest priority interrupts.

CAT2

In CAT2 type of interrupt where the ISR is handled by the system and OS calls can be called from the handler.

For CAT1 and CAT2, the selection of interrupt category is for each interrupt in the module. Individual MCAL module does not contain any option for interrupt category configuration. The user has to configure the ISR category in OS and to use the right MCAL specified name. MCAL expects “ISR(INTERRUPT_NAME)” keyword defined in OS for CAT2.

Notes:

1. OS module does not publish the short name handles for the CAT1 OsIsr container. However, it should be defined in the interrupt vector table by the OS.
2. OS module should publish the short name handles for the CAT2 OsIsr container according to ecuc_sws_2108 requirement by adding the Os_” prefix to the configured interrupt name.
3. As the file Interrupt_VectorTable.c provided is just a demo and not all interrupts will be mapped in this file, the user has to update the Interrupt_VectorTable.c according to their configuration.

4.2 Using MCAL integrated with Operation System:

<Msn> module's <Module>_Irq.c/h files include “Os.h” header file to obtain the interrupt category information configured in the OS. Therefore, the following pre-processor definitions are expected to be published in Os.h file by the OS for CAT2 or to be used in the interrupt vector table for CAT1.

Table 4-1 CAT1 and CAT2 Naming Convention

Interrupt Category	Naming Convention
CAT1	<MCAL_INTERRUPT_NAME>_ ISR
CAT2	<MCAL_INTERRUPT_NAME>_CAT2_ISR
CAT2 (In case the handles of the OsIsr container are generated without ‘Os_’ prefix by Os generation tool)	Os_<MCAL_INTERRUPT_NAME>_CAT2_ISR

4.3 Using MCAL stand alone

When using the MCAL modules standalone without the standard Autosar Os module, the user has to prepare an Os.h stub file with the published handles only for the interrupt names used as CAT2.

Table 4-2 List of ISR Names that need to be configured and published in Os.h (CAT2) or used in the interrupt vector table (CAT1) for <MSN> Driver

Sl. No.	CAT1	CAT2	CAT2(In case the handles of the OsIsr container are generated without ‘Os_’ prefix by Os generation tool)
1	<MSN>n_SGm_ISR	<MSN>n_SGm_CAT2_ISR	Os_<MSN>n_SGm_CAT2_ISR
2	<MSN>_DMA_CHxy_ISR	<MSN>_DMA_CHxy_CAT2_ISR	Os_<MSN>_DMA_CHxy_CAT2_IS R

Where

‘n’ indicates HW Unit number

‘m’ indicates SG Unit number

‘xy’ indicates DMA channel Id number

5.GNU MAKE Environment

Every component is delivered with the necessary Make scripts to integrate the component with the application. The scripts are compatible with GNU Make version 3.81.

All the delivered Makefiles have to be included in the project-level base Makefile in order to build the component together with the application. Refer to “R-Car Gen4 AUTOSAR R19-11 MCAL Modules Overview User’s Manual” 3.1 Driver Component Makefile for more information on the Makefile variables and their usage.

5.1 Build Process Without GNUMAKE

If GNU Make utility is not present at the default path or present in some other directory, the procedure below is followed to set the Environmental variable GNUMAKE\PATH.

1. Right click on “This PC” and select “Properties” to display the System window.

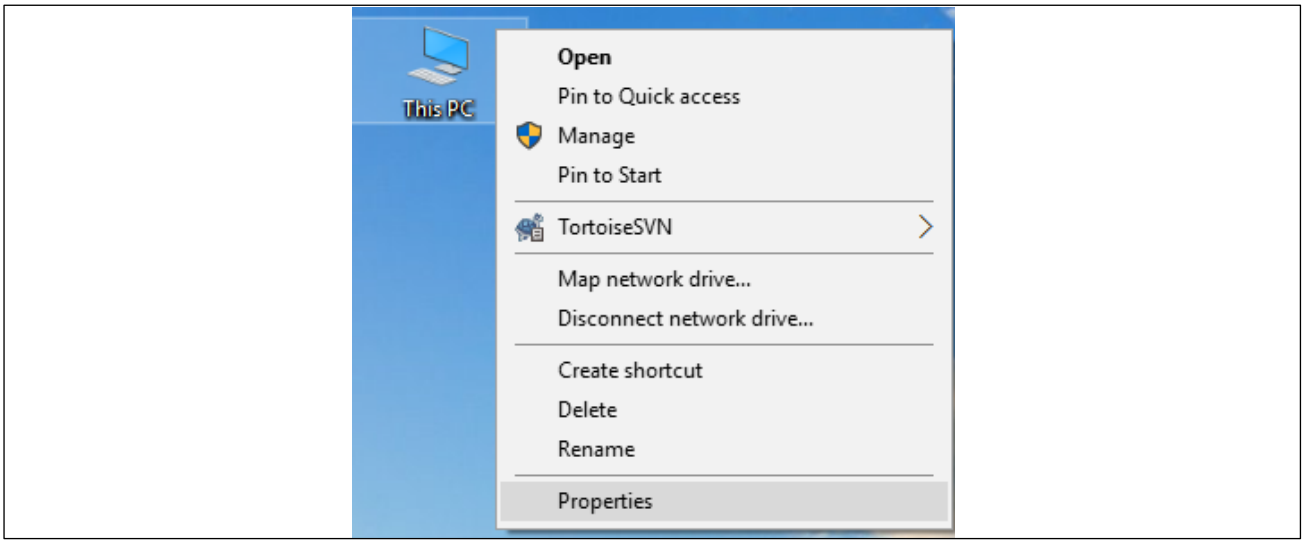


Figure 5.1 How to Open the System Window

2. In System window, click on “Advanced system settings” to display System Properties window.

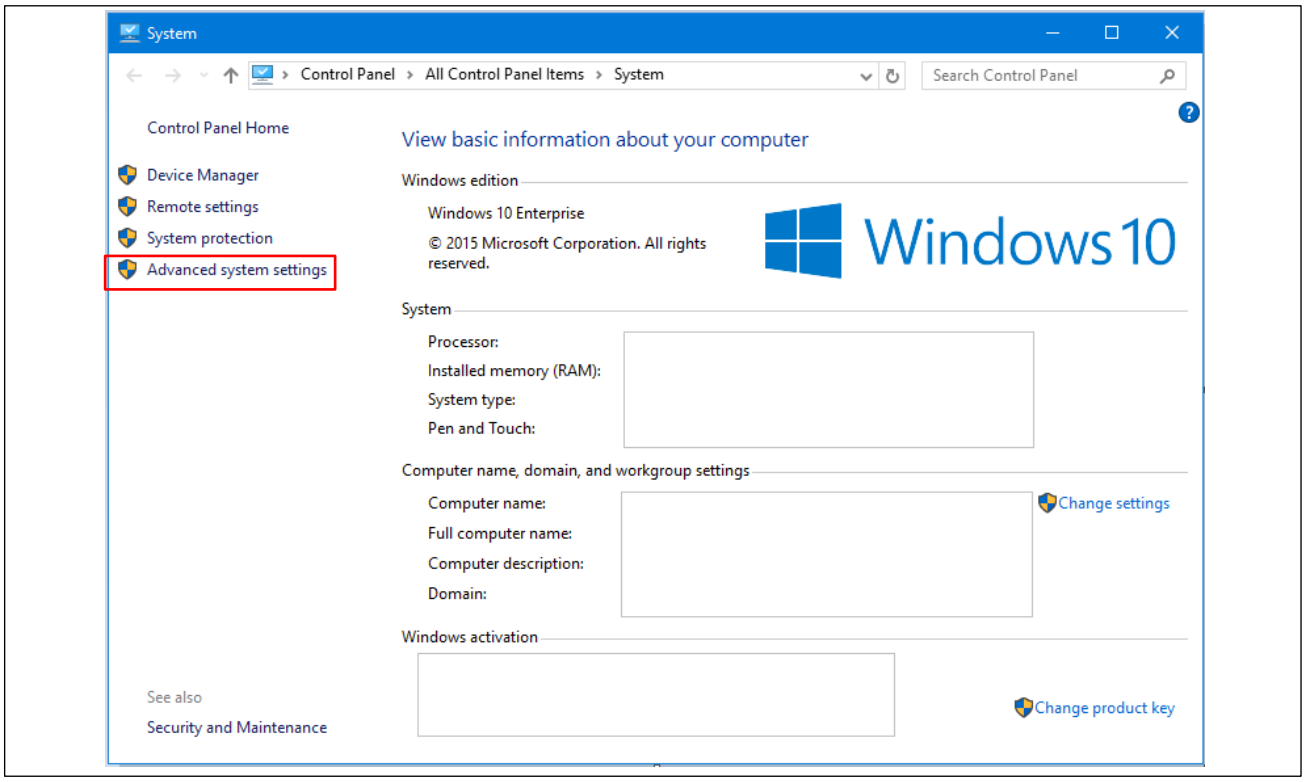


Figure 5.2 How to Open the Advanced System Setting Window

3. In System Properties window, select “Advanced” option to display “Environmental Variables” at the bottom of the window.

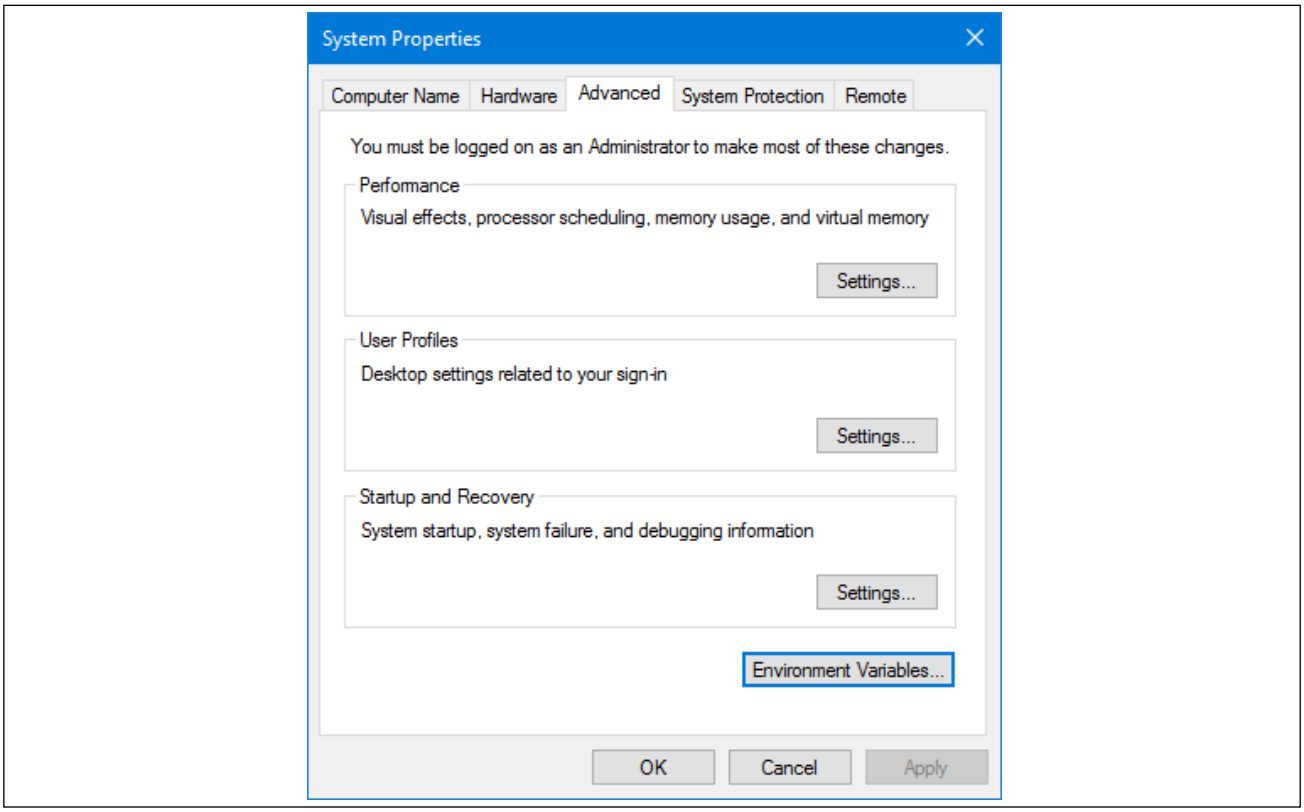


Figure 5.3 How to Open Environment Variables Window

4. Click on “Environmental Variables” to display “Environment Variables” window, then, select “New”.

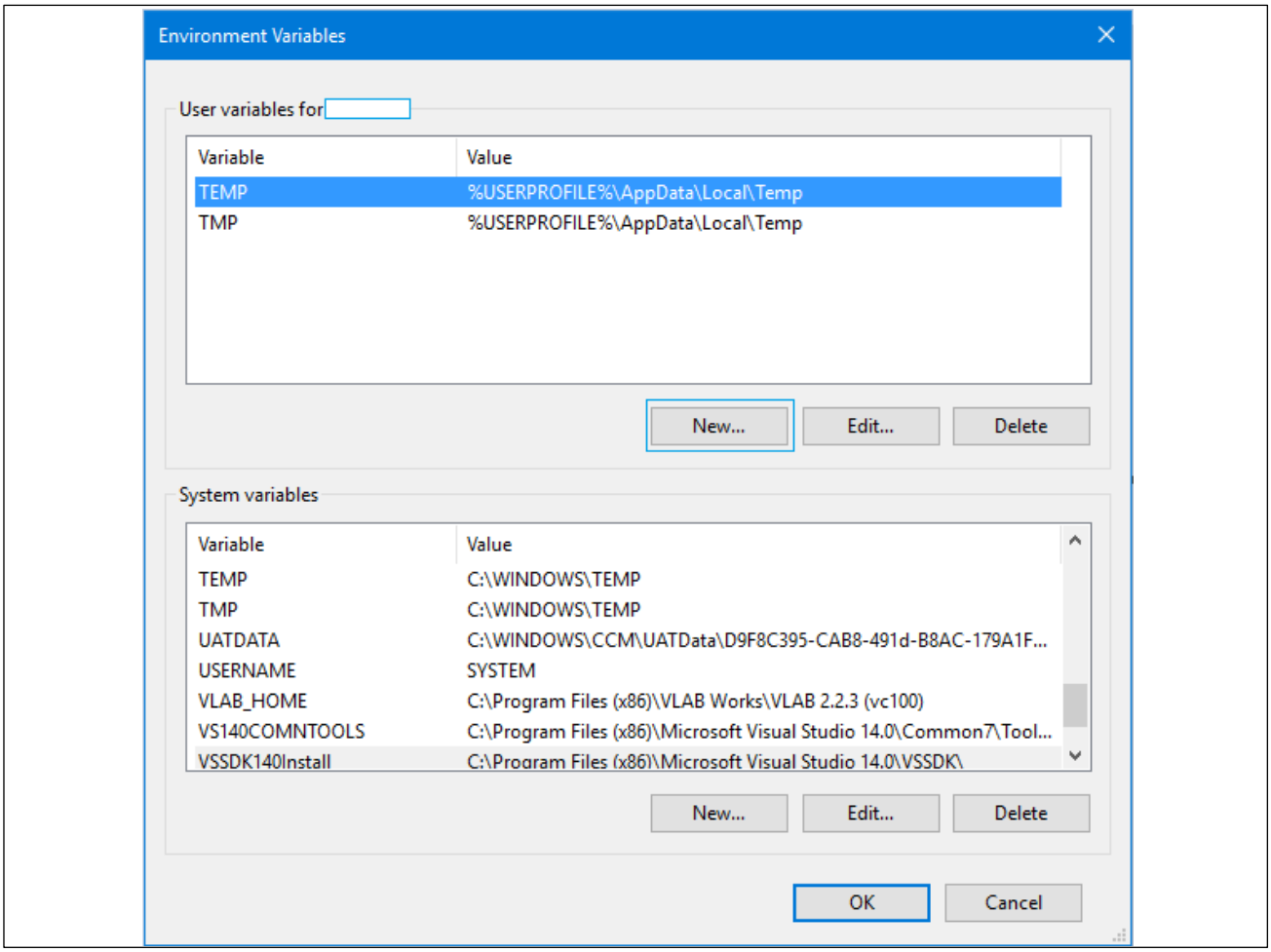


Figure 5.4 Environment Variables Window

5. After step 4, “New User Variable” window displays “Variable name” and “Variable value” options that needs to be set. Variable name will be set as GNUMAKE, and Variable value is the path of the directory where GNU Make utility is present. Click “OK”.

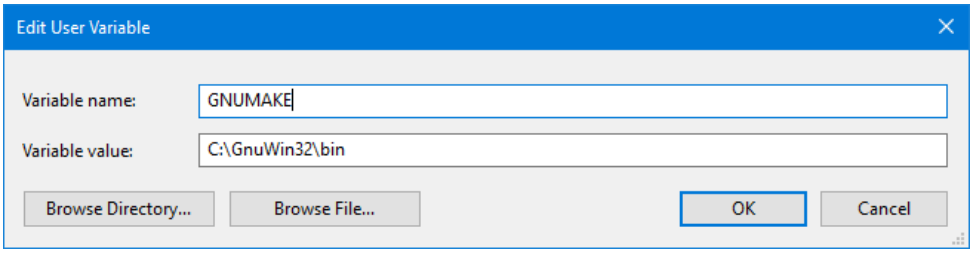


Figure 5.5 How to Set Environment Variables

6. After step 5, in “Environment Variables” window, click “OK”.
7. After step 6, in “System Properties” window, click “OK”.

Note GNU Make utility version 3.81 must be separately downloaded and installed to use the Makefiles delivered along with the component. For more information on the utility, visit http://www.gnu.org/

5.2 Build Process With GNUMAKE

When the batch file of certain application is built, the GNU Make utility will be searched by batch file. The GNU Make utility should be present in the default path specified by GNUMAKE\PATH variable. The batch file is compiled by using the GNU Make utility.

6.Load Binaries

Once the Executable or S-Record is generated using the project-level base Makefile, it needs to be downloaded to the target using a Flash programmer.

The user must read the instructions provided in the Flash programmer’s User Manual thoroughly before using it.

Revision History		R-Car Gen4 AUTOSAR R19-11 MCAL User's Manual	
Rev.	**Date**	**Description**	
		Page	**Summary**
1.00	Oct 08, 2021	—	First Edition issued
1.01	Nov 30, 2021	—	Chapter 2: - Add Note of FLS, ICCOM, IIC for Perl Generation Tool
1.02	Dec 17, 2021	—	Update Publication Date
1.03	Jan 28, 2022	—	Update Publication Date
1.04	Apr 25, 2022	—	Update Publication Date Chapter 2: - Add Note of EMM, IPMMU, THS, RFSO, CRC for Perl Generation Tool and the other modules for C Sharp Generation tool
1.05	May 27, 2022	—	Update Publication Date
1.06	July 27, 2022	—	Update Publication Date Chapter 2: - Remove Note EMM, THS, IIC, FLS, RFSO, CRC, IPMMU in for Perl Generation Tool. And add Note for EMM, THS, IIC, FLS, RFSO, CRC, IPMMU in C Sharp Generation tool.
1.07	Nov 17,2022	—	Update Publication Date Chapter 2: Section 2.1: Add Files that deviate from the naming conventions Section 2.3: Remove the description related to multi-instance. Chapter 3: Section 3.2: Remove the description related to multi-instance. Chapter 4: Add section name 4.1, 4.2, 4.3 to make clear the relation Interrupt Categories.
2.00	Jan 17, 2023	—	Update Publication Date
2.01	Oct 27, 2023	—	- Add V4M attribute and device for all requirement. Chapter 2: - Add V4M to not supported device at section 2.3.

R-Car Gen4 AUTOSAR R19-11 MCAL User's Manual
Getting Started

Publication Date: Rev.2.01 Oct 27, 2023

Published by: Renesas Electronics Corporation

R-Car Gen4 AUTOSAR R19-11 MCAL
User's Manual

Renesas Electronics Corporation